DEEP LEARNING COM TENSORFLOW

REPRESENTATION & TRANSFER LEARNING

DIEGO RODRIGUES DSC

INFNET

CRONOGRAMA

Dia	Aula	Trab
~~02/09~~	~~Workshop de Deep Learning~~	
~~04/09~~	~~Deep FeedForward~~	
~~09/09~~	~~Rede Neural Convolutiva~~	~~Modelo Baseline~~
~~11/09~~	~~AutoEncoders~~	
16/09	Representation & Transfer Learning	Modelo Profundo
18/09	Sequências	
23/09	Modelos Generativos	Deployment
25/09	Apresentação dos Trabalhos Parte II	

REPRESENTATION LEARNING & TRANSFER LEARNING

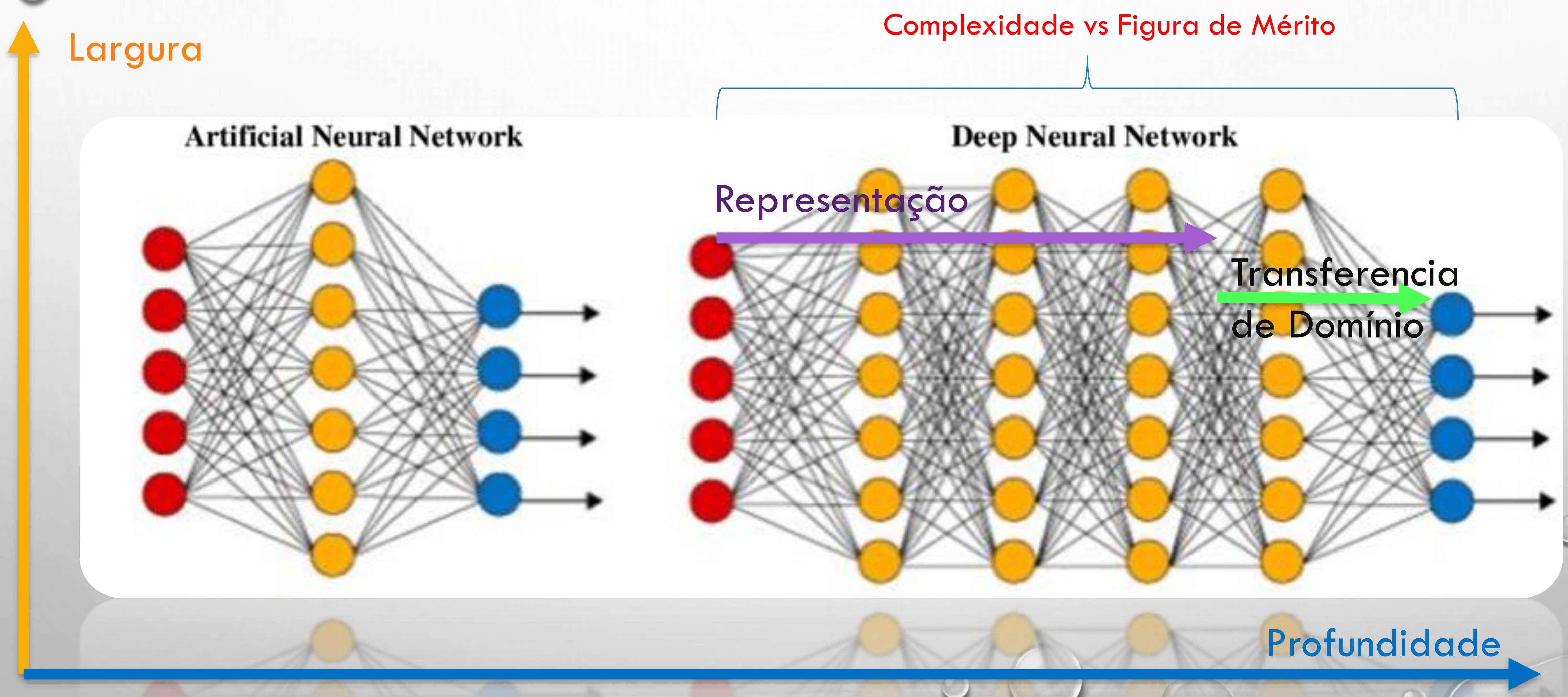
- PARTE 1 : TEORIA
  - MODELING
    - ESPAÇO DE REPRESENTAÇÕES
    - APRENDIZADO DE REPRESENTAÇÃO
    - TRANSFER LEARNING
    - TRANSFERENCIA DE DOMÍNIO

- PARTE 2 : PRÁTICA
  - TRANSFER LEARNING
- PARTE 3 : TRABALHOS

PARTE 1 : TEORIA

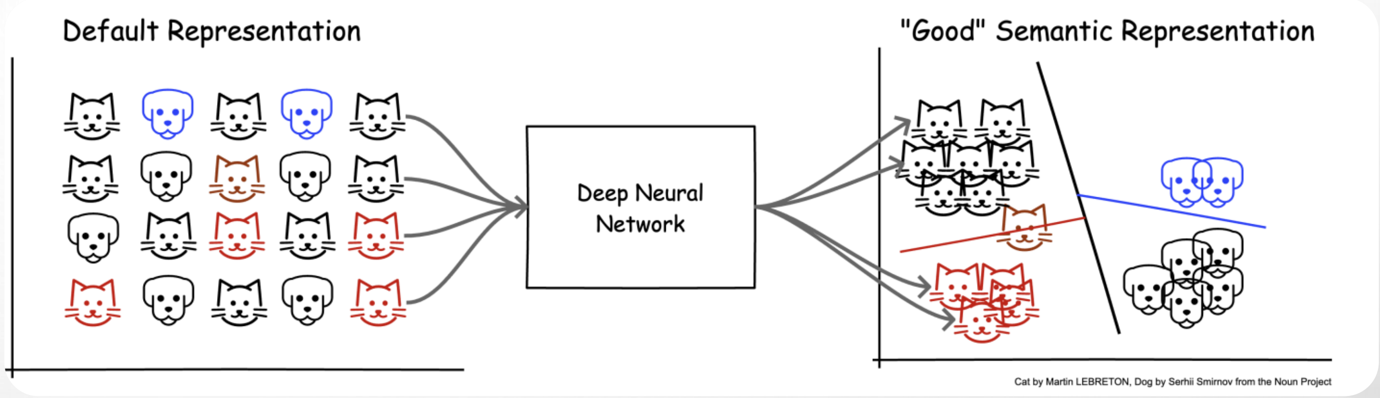
MODELING

ESPAÇO DE REPRESENTAÇÕES



APRENDIZADO DE REPRESENTAÇÕES

O aprendizado de representações se refere à capacidade dos modelos profundos de automaticamente identificar e extrair características relevantes dos dados.

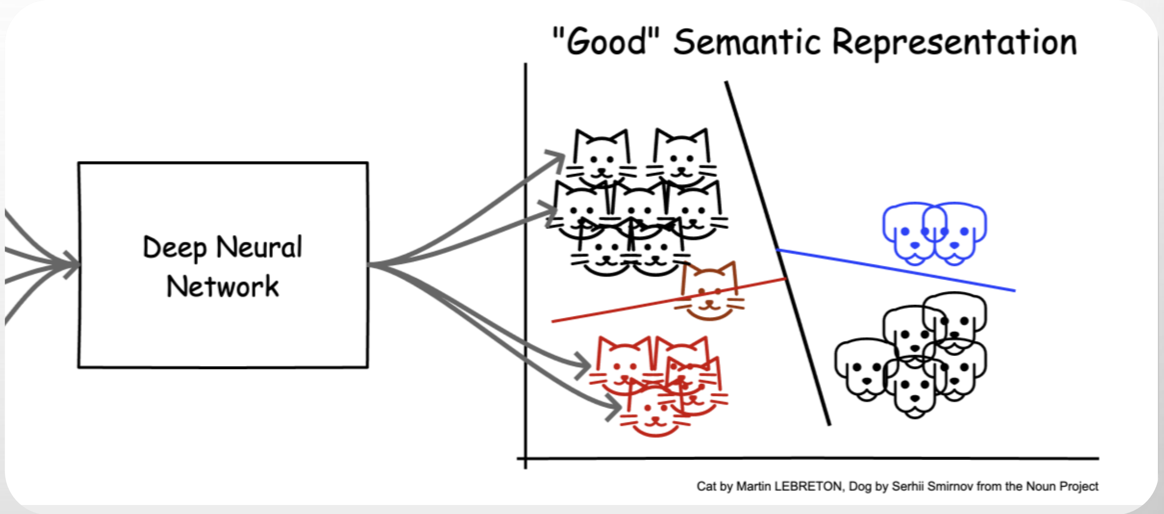


Em vez de depender de **feature engineering,** o objetivo do **Aprendizado de Representações** é permitir que o modelo construa suas próprias representações a partir de dados brutos.

Essas representações geralmente têm a forma de **camadas intermediárias de redes neurais,** que capturam **abstrações ou padrões úteis.**

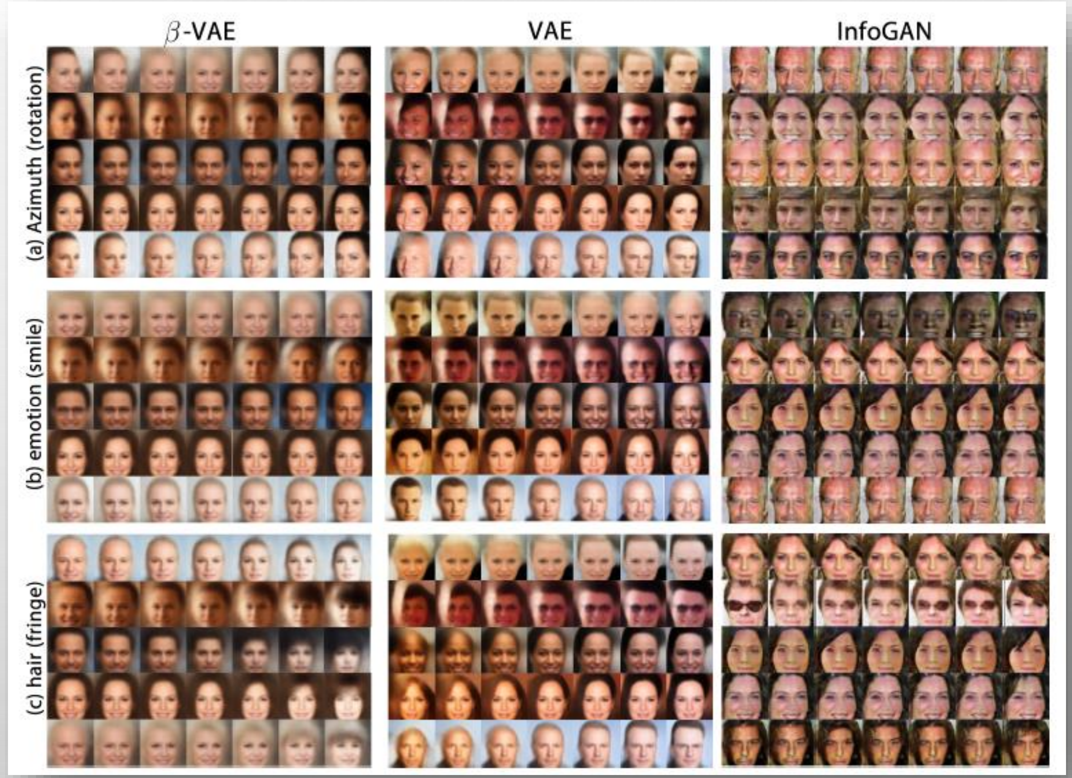
MÉTODOS DE APRENDIZADO DE REPRESENTAÇÃO

1. **Autoencoders:** Para compressão e aprendizado de representações latentes.
2. **Modelos Pré-treinados:** Para capturar representações ricas em tarefas amplas e adaptá-las para novas tarefas.
3. **Redes Convolutivas (CNNs):** Para aprendizado de representações hierárquicas em imagens.
4. **Modelos Sequenciais (RNNs, LSTMs, GRUs):** Para aprendizado de representações de sequências temporais ou dados com dependências de ordem.



REPRESENTAÇÃO VIA AUTOENCODERS

Autoencoders: Para compressão e aprendizado de representações latentes.



REPRESENTAÇÃO VIA MODELOS PRÉ-TREINADOS

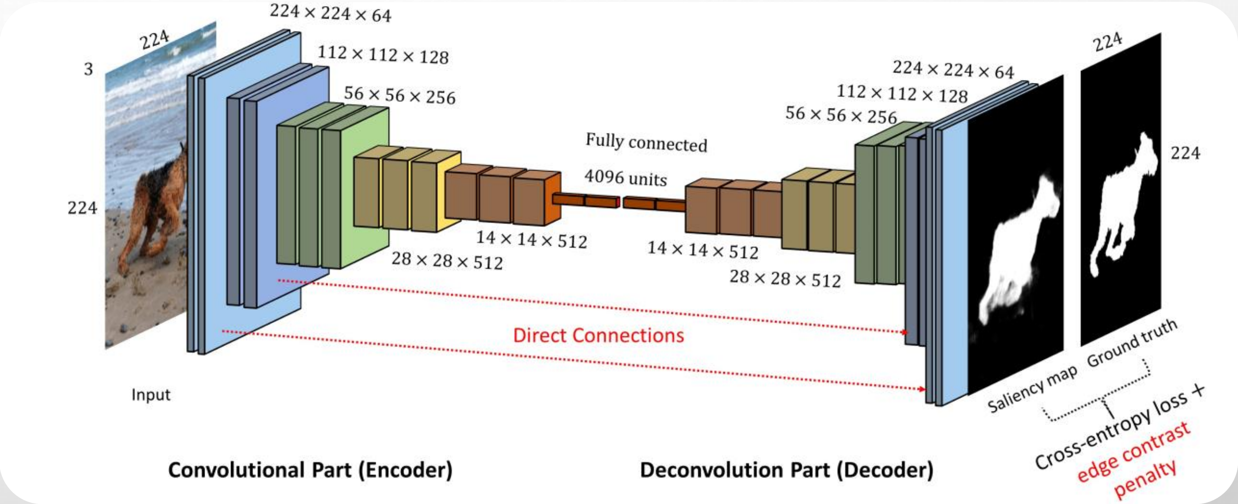
Modelos Pré-treinados: Para capturar representações ricas em tarefas amplas e adaptá-las para novas tarefas.

Available models

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	79.0%	94.5%	22.9M	81	109.4	8.1
VGG16	528	71.3%	90.1%	138.4M	16	69.5	4.2
VGG19	549	71.3%	90.0%	143.7M	19	84.8	4.4
ResNet50	98	74.9%	92.1%	25.6M	107	58.2	4.6
ResNet50V2	98	76.0%	93.0%	25.6M	103	45.6	4.4
ResNet101	171	76.4%	92.8%	44.7M	209	89.6	5.2
ResNet101V2	171	77.2%	93.8%	44.7M	205	72.7	5.4
ResNet152	232	76.6%	93.1%	60.4M	311	127.4	6.5
ResNet152V2	232	78.0%	94.2%	60.4M	307	107.5	6.6
InceptionV3	92	77.9%	93.7%	23.9M	189	42.2	6.9
InceptionResNetV2	215	80.3%	95.3%	55.9M	449	130.2	10.0
MobileNet	16	70.4%	89.5%	4.3M	55	22.6	3.4
MobileNetV2	14	71.3%	90.1%	3.5M	105	25.9	3.8
DenseNet121	33	75.0%	92.3%	8.1M	242	77.1	5.4
DenseNet169	57	76.2%	93.2%	14.3M	338	96.4	6.3
DenseNet201	80	77.3%	93.6%	20.2M	402	127.2	6.7
NASNetMobile	23	74.4%	91.9%	5.3M	389	27.0	6.7
NASNetLarge	343	82.5%	96.0%	88.9M	533	344.5	20.0
EfficientNetB0	29	77.1%	93.3%	5.3M	132	46.0	4.9
EfficientNetB1	31	79.1%	94.4%	7.9M	186	60.2	5.6
EfficientNetB2	36	80.1%	94.9%	9.2M	186	80.8	6.5
EfficientNetB3	48	81.6%	95.7%	12.3M	210	140.0	8.8
EfficientNetB4	75	82.9%	96.4%	19.5M	258	308.3	15.1
EfficientNetB5	118	83.6%	96.7%	30.6M	312	579.2	25.3
EfficientNetB6	166	84.0%	96.8%	43.3M	360	958.1	40.4

REPRESENTAÇÃO REDE NEURAL CONVOLUTIVA

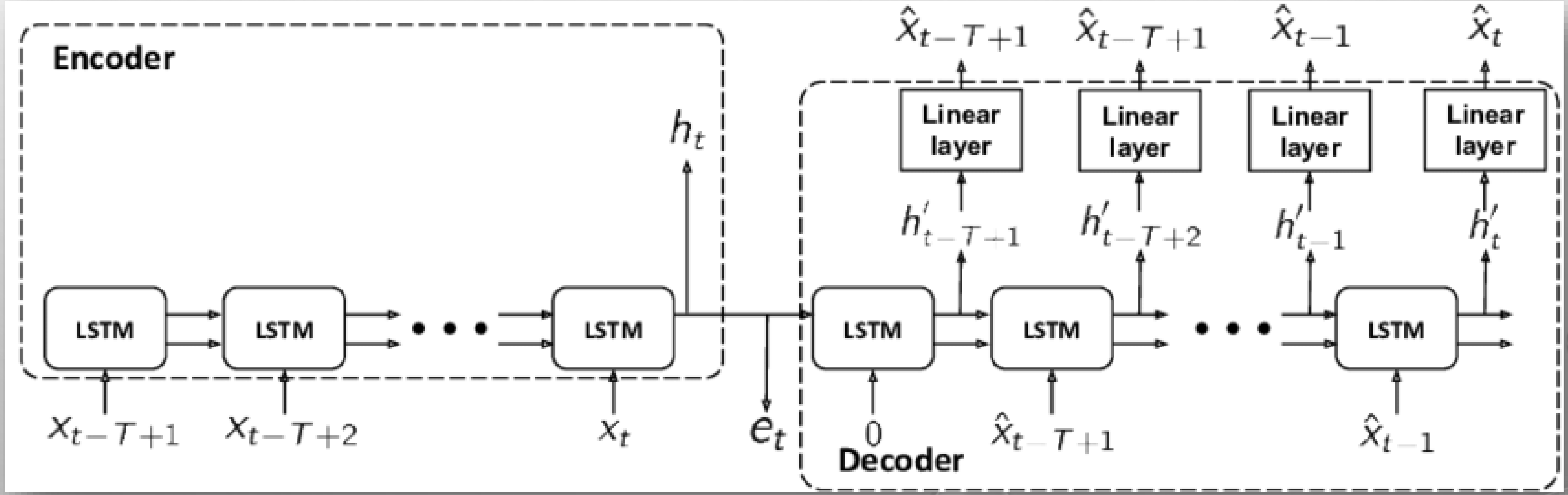
Redes convolutivas (CNNs): Para aprendizado de representações hierárquicas em imagens.



Masci, J., Meier, U., Ciresan, D., & Schmidhuber, J. (2011). "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction". *International Conference on Artificial Neural Networks (ICANN)*

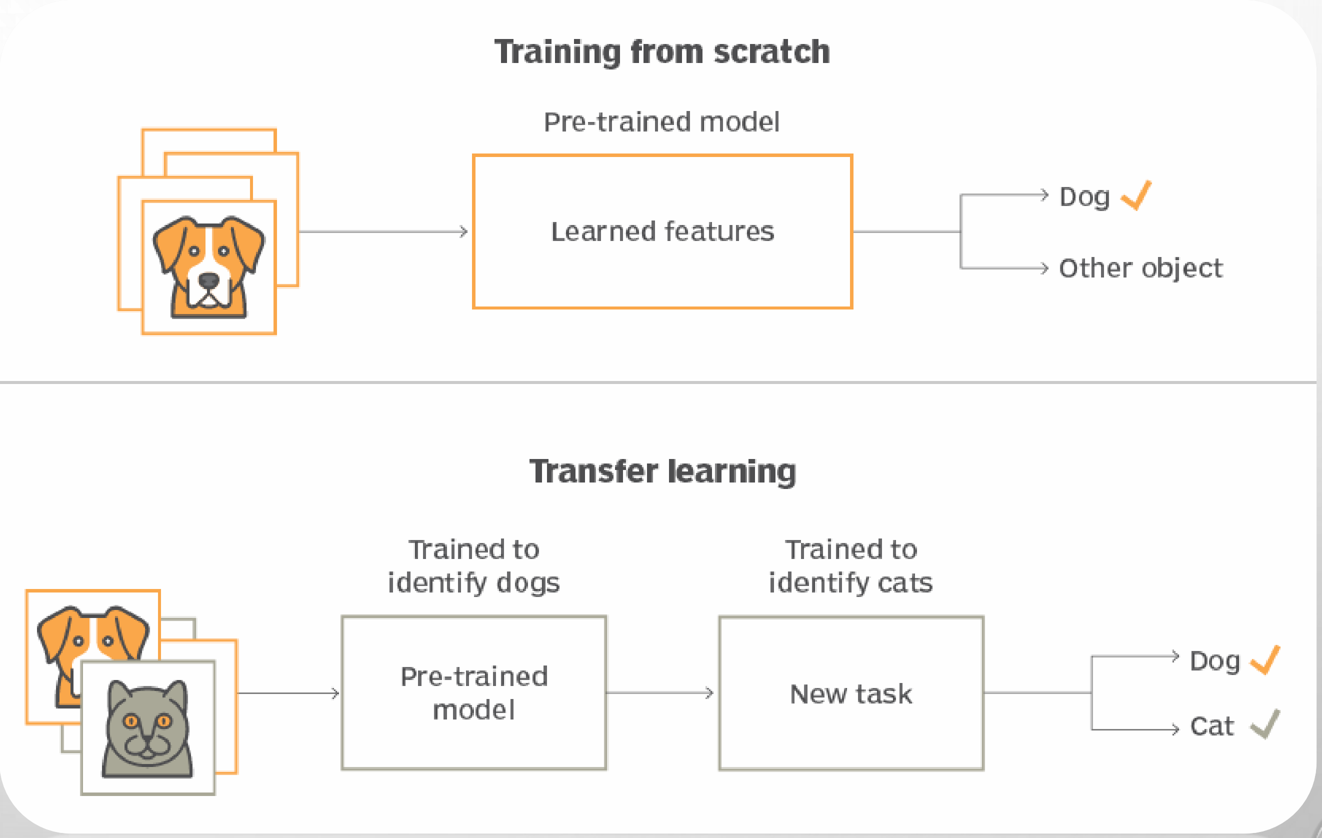
REPRESENTAÇÃO POR MODELOS SEQUENCIAIS

Modelos Sequenciais (RNNs, LSTMs, GRUs): Para aprendizado de representações de sequências temporais ou dados com dependências de ordem.



TRANSFER LEARNING

Transfer Learning é a técnica onde um modelo treinado em uma **tarefa (domínio de origem)** é reutilizado em uma nova tarefa **(domínio de destino).** O conhecimento aprendido em uma tarefa anterior é aplicado para **acelerar o aprendizado em uma nova tarefa**, que pode ser similar ou diferente.



TRANSFERENCIA DE DOMINIO

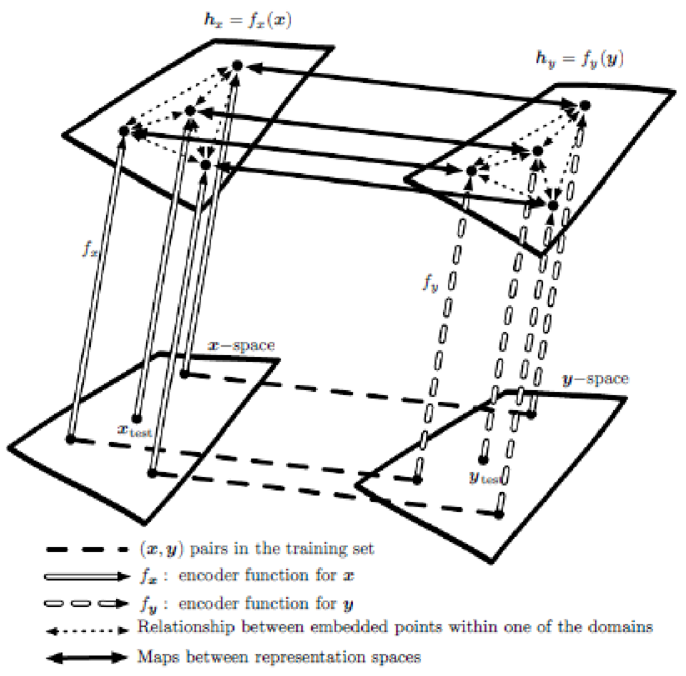


Figure 15.3: Transfer learning between two domains $\boldsymbol{x}$ and $\boldsymbol{y}$ enables zero-shot learning. Labeled or unlabeled examples of $\boldsymbol{x}$ allow one to learn a representation function f_x and similarly with examples of $\boldsymbol{y}$ to learn f_y. Each application of the f_x and f_y functions appears as an upward arrow, with the style of the arrows indicating which function is applied. Distance in $\boldsymbol{h}_x$ space provides a similarity metric between any pair of points in $\boldsymbol{x}$ space that may be more meaningful than distance in $\boldsymbol{x}$ space. Likewise, distance in $\boldsymbol{h}_y$ space provides a similarity metric between any pair of points in $\boldsymbol{y}$ space. Both of these similarity functions are indicated with dotted bidirectional arrows. Labeled examples (dashed horizontal lines) are pairs $(\boldsymbol{x}, \boldsymbol{y})$ which allow one to learn a one-way or two-way map (solid bidirectional arrow) between the representations $f_x(\boldsymbol{x})$ and the representations $f_y(\boldsymbol{y})$ and anchor these representations to each other. Zero-data learning is then enabled as follows. One can associate an image $\boldsymbol{x}_{\text{test}}$ to a word $\boldsymbol{y}_{\text{test}}$, even if no image of that word was ever presented, simply because word-representations $f_y(\boldsymbol{y}_{\text{test}})$ and image-representations $f_x(\boldsymbol{x}_{\text{test}})$ can be related to each other via the maps between representation spaces. It works because, although that image and that word were never paired, their respective feature vectors $f_x(\boldsymbol{x}_{\text{test}})$ and $f_y(\boldsymbol{y}_{\text{test}})$ have been related to each other. Figure inspired from suggestion by Hrant Khachatrian.

KERAS TRANSFER LEARNING

► Developer guides / Transfer learning & fine-tuning

Transfer learning & fine-tuning

Author: fchollet
Date created: 2020/04/15
Last modified: 2023/06/25
Description: Complete guide to transfer learning & fine-tuning in Keras.

View in Colab • GitHub source

Setup

```
import numpy as np
import keras
from keras import layers
import tensorflow_datasets as tfds
import matplotlib.pyplot as plt
```

Introduction

Transfer learning consists of taking features learned on one problem, and leveraging them on a new, similar problem. For instance, features from a model that has learned to identify racoons may be useful to kick-start a model meant to identify tanukis.

Transfer learning is usually done for tasks where your dataset has too little data to train a full-scale model from scratch.

The most common incarnation of transfer learning in the context of deep learning is the following workflow:

PARTE 2 : PRÁTICA

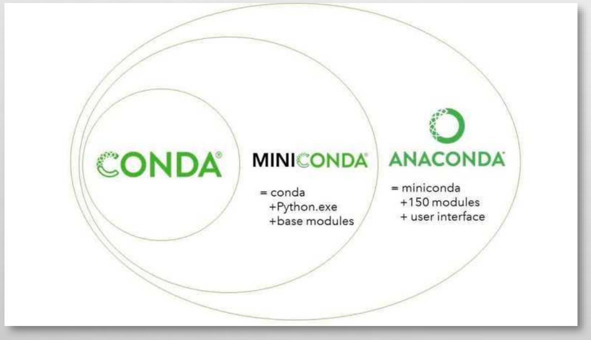
AMBIENTE PYTHON

6. Machine Learning

6. Deployment

5. Visualização

4. Variáveis Aleatórias



1. Editor de Código

2. Gestor de Ambiente

3. Ambiente Python do Projeto

3. Notebook Dinâmico

WORKSHOP

- ~~QUAL A TOPOLOGIA DE DEEP LEARNING ADEQUADA PARA O MEU TRABALHO?~~
- ~~QUAL CAPÍTULO DO LIVRO MELHOR SE ENQUADRA NO MEU TRABALHO?~~
- ~~AULA 3: NOVO CICLO DE BUSINESS UNDERSTANDING / GRUPO + MODELO BASELINE TREINADO~~
- **AULA 5 OU 7: MODELO PROFUNDO TREINADO**
- **AULA 7: DEPLOYMENT DO MODELO***
- **AULA 3-7 > APRESENTAÇÃO TEÓRICA DA(S) TOPOLOGIA(S) + LEITURA DE ARTIGO + ACOMPANHAMENTO DOS TRABALHOS + DEEP DIVE NO CÓDIGO (POR GRUPO)**
- APRESENTAÇÃO FINAL DOS TRABALHOS

**Passo Opcional*